

Universidad de las Ciencias Informáticas

Facultad 5



Título: Evaluación de la Eficiencia de componentes de software.

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autora:

Tania Vega Rodríguez.

Tutoras:

M Sc. Liudmila Reyes Álvarez.

Ing. Yadira Morales Álamo.

Co-Tutora:

Ing. Alicia Lisett Fernández

“La responsabilidad nuestra es luchar porque la calidad del producto que aquí se haga sea de las mejores y la mejor posible.”

Ernesto “Che” Guevara

Declaración de autoría

Declaro ser autora de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año_____.

Firma de la autora
Tania Vega Rodríguez

Firma de la Tutora
M Sc. Liudmila Reyes Álvarez

Firma de la Tutora
Ing. Yadira Morales Álamo

Agradecimientos

A mis padres, en especial a mi papá que fue, es y será lo más importante en mi vida, gracias por estar siempre en el momento justo, con el consejo oportuno, con la ayuda incondicional, con el sacrificio sin límites...

A Yaquelín y a Vanesa, por ser una madre y una hermana para mí, porque mi vida tuvo un antes y un después de ustedes, muchas gracias.

A Tata, por estar conmigo siempre, en los momentos de alegría, pero sobre en los más difíciles, porque la he tenido en cada paso de mi vida desde que vi la luz por primera vez hasta el día de hoy.

A mis amistades de siempre y para siempre, en especial a Yeny, Dayana, Yanet, Marlén y Ernesto por los momentos vividos que son inolvidables.

A las instructoras Daisy y Mercedes, por enseñarnos cada día, por aguantarnos las malcriadeces y por el ejemplo siempre.

A las tutoras Liudmila y Yadira, por la guía certera en la realización de la investigación.

Dedicatoria:

Este trabajo de diploma está dedicado a las personas más importantes de mi vida: papi, mamiti, titina y tata, están y estarán siempre en lo más alto de mi pirámide, por estar siempre ahí para mí, muchas gracias.

Resumen:

La oferta de software evoluciona de forma creciente en el mercado, igualmente la demanda de dichos sistemas es cada vez mayor. Por lo que se hace necesario producirlos con la mejor calidad posible siempre en concordancia con los requisitos funcionales y de rendimiento, explícitamente establecidos con los estándares de desarrollo correctamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente. Además para asegurar la calidad de los mismos se requiere de un conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza en que el producto (software) satisfará los requisitos dados de calidad, para poder de esta manera competir en el mercado. Es por ello que en el siguiente trabajo de diploma se elabora un procedimiento que permitirá evaluar la eficiencia de los componentes de software desarrollados en el CEDIN (Centro de Desarrollo de Informática Industrial). Este procedimiento se valida en dos componentes, HMI (Human Machine Interface) y Seguridad, del proyecto Energía UCI del SCADA-UX.

Palabras clave: software, requisitos, calidad, eficiencia, procedimiento.

Índice

INTRODUCCIÓN	11
CAPÍTULO1 FUNDAMENTACIÓN TEÓRICA	15
Introducción	15
1.1 Desarrollo de software basado en componentes.....	15
1.1.1 Características claves para que un elemento sea catalogado como componente.....	16
1.1.2 Tipos de componentes.	16
1.2 Calidad de un Producto Software.....	17
1.3 Eficiencia: Estado del arte.	19
1.1.3 Otras definiciones de eficiencia.	20
1.1.4 Sub-Características.	21
1.1.5 Diferencias con la eficacia.....	21
1.1.6 Herramientas para medir eficiencia.....	22
1.1.7 Las métricas para la medición de la característica eficiencia.....	26
1.4 Definición de procedimiento.....	26
Conclusiones Parciales.....	26
CAPÍTULO 2 SOLUCIÓN PROPUESTA	28
Introducción:	28
2.1 Nombre del procedimiento	29
2.2 Objetivos	29
2.3 Alcance.....	29
2.4 Referencias.....	30
2.5 Responsables.....	30
2.6 Términos y definiciones.....	30
2.7 Normas generales.....	30
2.8 Etapas del procedimiento.....	31
2.8.1 Etapa1: Caracterización del componente a evaluar en cualquier fase de desarrollo.....	31
2.8.2 Etapa2: Descripción de la evaluación.....	35
2.8.3 Etapa3: Ejecución de la medición de la eficiencia.....	41
2.8.4 Etapa4: Análisis de la información obtenida de la medición.....	43
Conclusiones Parciales	45
CAPÍTULO 3 VALIDACIÓN DEL PROCEDIMIENTO.....	46
Introducción:	46
3.1 Descripción del procedimiento.....	46
3.1.1 Componente HMI.....	47
3.1.2 Componente Seguridad.....	50
3.2 Comparación entre los componentes:.....	52
Conclusiones Parciales:.....	53

CONCLUSIONES GENERALES:	54
RECOMENDACIONES	55
REFERENCIAS BIBLIOGRÁFICAS:	56
BIBLIOGRAFÍA:	57
ANEXOS	58
Anexo 1: Informe de Evaluación de la Eficiencia	58

INDICE DE ILUSTRACIONES

Ilustración 1: Etapas del procedimiento.....	29
Ilustración 2: Actividades para la etapa 1	31
Ilustración 3: Actividades para la etapa2.....	36
Ilustración 4: Actividades de la etapa3.....	41
Ilustración 5: Actividades para la etapa4.....	43
Ilustración 6: Vista ejecución de HMI	48
Ilustración 7: Vista edición de HMI	48
Ilustración 8: Resultado del procedimiento para HMI	52
Ilustración 9: Resultado del procedimiento para Seguridad	52

INDICE DE TABLAS

Tabla 1: Correspondencia de las sub-características por estado fase y rol	35
Tabla 2. Métricas por sub-características.....	39
Tabla 3: Niveles de eficiencia.....	41
Tabla 5: Resultado de las sub-características.....	42
Tabla 4: Resultado de las métricas	42
Tabla 6: Errores y acciones correctivas	44
Tabla 7: Errores comunes y sus correspondientes acciones correctivas	44
Tabla 8: Métricas que se aplican por sub-características.....	47
Tabla 9: Resultado de la aplicación del procedimiento a HMI	50
Tabla 10: Niveles por sub-características para HMI.....	50
Tabla 11: Niveles de rendimiento para Seguridad.....	51

INTRODUCCIÓN

El mundo en el que hoy vivimos es cada vez más dependiente del software, y las necesidades humanas en todos los sectores, tributan exponencialmente a su desarrollo, sobre todo en materia de economía digital, evolución de empresas y conocimiento. Sin embargo se hace necesario desarrollar productos que cumplan con los requisitos, se ajusten a los planes y a los presupuestos asignados, factores de vital importancia para cualquier empresa productora de software, cuya significación en el desarrollo de productos informáticos crece vertiginosamente, convirtiéndose en un elemento imprescindible en el desarrollo de esta industria.

La gran competencia entre las organizaciones que se desenvuelven en esta rama a nivel mundial exige que se deba asegurar que el proceso de desarrollo de software esté dirigido a perfeccionar el nivel de satisfacción del cliente como resultado de las actividades, lo que implica mejorar la calidad y la eficiencia del producto final.

Cuba todavía no ha alcanzado grandes resultados en materia de software sin embargo no está ajeno a ello, entre las tareas priorizadas del gobierno cubano se encuentra potenciar esta industria, aspirando a alcanzar resultados lo más rápido posible, para ello es necesario insertarse en el mercado mundial con productos de alta calidad en cuanto a una serie de características entre ellas la eficiencia, que impulsarán un nuevo paradigma productivo: Empresa de Software, respondiendo a crecientes demandas del mercado Latinoamericano y Europeo. Para lograrlo con éxito hay que tener en cuenta aspectos importantes entre los que se encuentran: la gestión del proyecto, el desarrollo técnico, los sistemas de calidad, incluyendo las actividades de validación y el sistema de gestión de configuración de software.

La Universidad de la Ciencias Informáticas es una muestra fehaciente de la inserción de Cuba en el camino hacia el universo del software. Hoy después de 8 años de fundada se pueden observar algunos logros, teniendo siempre como premisa lograr la mayor calidad del producto final y como prioridad la plena satisfacción del cliente.

El centro de Desarrollo de Informática Industrial (CEDIN) aplica como el resto de la universidad el Desarrollo de Software Basado en Componentes (DSBC). Para garantizar la calidad del producto final, que es conformado por varios componentes, se requiere que todos los procesos y artefactos que son resultados del ciclo de vida del software tengan la calidad adecuada, de esta manera se debe garantizar que los componentes que se van desarrollando a partir de estos

respondan a los estándares y modelos de calidad aplicados en la universidad. Los componentes que actualmente se han ido desarrollando en el CEDIN evalúan la eficiencia sin el uso de un procedimiento, por lo que la forma de conocer el grado de eficiencia se hace muy subjetiva y los resultados pueden ser insuficientes comprometiendo de esta forma la calidad del componente que se desarrolla y con ello la calidad del producto final. Medir la eficiencia de los componentes es de suma importancia para el centro pues es a partir de la misma que se puede conocer el grado de rendimiento y utilización de los recursos sin este dato se trabaja a ciegas, o sea, se desconoce cuán bien o cuán mal se está con respecto a la competencia, además que si el componente es liberado y utilizado en un producto X, pero no es lo suficientemente eficiente se ve comprometida la calidad del producto. Esta situación pone en riesgo la satisfacción del cliente y por consiguiente el prestigio de la universidad.

La **situación problemática** antes expuesta plantea la necesidad de dar respuesta al siguiente **problema científico**:

- ¿Cómo evaluar un componente software de acuerdo a su Eficiencia en el Centro de Informática Industrial?

El **objeto de investigación** del trabajo es:

- Evaluación de la calidad de software basado en componentes.

La investigación tiene como **objetivo general**:

- Elaborar un procedimiento que permita evaluar la Eficiencia de componentes de software que se desarrollan en el CEDIN.

A partir de este se puede determinar como **campo de acción**:

- Evaluación de la calidad de un componente de software en cuanto a su eficiencia.

La investigación tiene como **idea a defender**:

- Si se elabora un procedimiento para evaluar la calidad del software en cuanto a la eficiencia se debe contribuir a garantizar que todos los componentes de software que se liberen en

el CEDIN se les pueda medir el grado de eficiencia y por consiguiente de haber algún error con ellos se pueda remediar a tiempo o al menos registrar su presencia.

Para alcanzar el objetivo planteado se trazan las siguientes **tareas investigativas**:

- Elaboración del marco teórico a través del estudio del estado del arte.
- Descripción de los principales estándares o modelos de calidad con el objetivo de determinar el que más se adecue a la solución del procedimiento.
- Caracterización de las distintas técnicas y conceptos de medición y evaluación de la calidad de software en cuanto a la eficiencia para un mejor entendimiento de la problemática.
- Valoración de la calidad de los componentes de software para definir cómo se podrá evaluar la calidad de un componente en cuanto a su Eficiencia en el Centro de Informática Industrial (CEDIN).
- Confección del procedimiento para evaluar la calidad de los componentes en cuanto a la Eficiencia de los mismos, en el Centro de Informática Industrial.
- Validación del procedimiento por el método de caso de estudio en algún componente del CEDIN.

Entre los métodos de trabajo científico utilizados se destacan los siguientes:

Teóricos:

-El método histórico-lógico para el estudio crítico de los trabajos anteriores, y para utilizar estos como punto de referencia y comparación de los resultados alcanzados.

-El método analítico-sintético al descomponer el problema de investigación en elementos por separado y profundizar en el estudio de cada uno de ellos, para luego sintetizarlos en la solución de la propuesta

Empíricos:

-El método estadístico para realizar un estudio de las métricas a utilizar en la elaboración del procedimiento.

La presente investigación tiene la siguiente estructura de capítulos:

Capítulo 1: “Fundamentación Teórica”, que aborda el marco teórico de la investigación y el análisis de las normas”. En este capítulo se exponen una serie de definiciones y conceptos que ayudarán a comprender el tema, entre los conceptos que podemos encontrar están el de eficiencia, calidad de productos software, la diferencia entre eficiencia y eficacia, entre otros.

Capítulo 2: “Elaboración del procedimiento”. Se elabora el procedimiento que permitirá medir el grado de eficiencia de cualquier componente software del CEDIN.

Capítulo 3: “Validación del procedimiento”. Donde se realiza la validación del procedimiento en los componentes HMI y Seguridad pertenecientes al proyecto Energía UCI del Scada-ux.

Capítulo 1 *Fundamentación teórica*

Introducción

Teniendo en cuenta que este trabajo de diploma debe contribuir metodológicamente a evaluar la eficiencia del software basado en componentes, se hace necesario conocer primeramente los conceptos de calidad de un producto de software, así como una breve descripción de cada una de las características que miden la calidad de un producto o componente software, para luego profundizar en la característica de eficiencia a la que tributa esta investigación, así como a cada una de sus sub-características, lo cual se conocerá a través del estudio del estado del arte, del concepto de eficiencia, así como las herramientas y técnicas para evaluarla. Este capítulo estará estructurado en epígrafes que profundizarán en los aspectos antes mencionados y finalmente aparecerán las conclusiones parciales.

1.1 Desarrollo de software basado en componentes.

El desarrollo de software basado en componentes (DSBC), es una tecnología que ofrece ventajas en tiempo de desarrollo y reducción de costos en el proceso de desarrollo de software.

Los desarrollos tradicionales de aplicaciones incurren en altos costos y en una inversión de tiempo extensa. El iniciar un desarrollo de software desde cero es un reto muy grande, incluso para una empresa que pueda soportar este proceso. El DSBC busca, dentro de otros objetivos, reducir el tiempo de trabajo, el esfuerzo que requiere implementar una aplicación y los costos del proyecto, y, de esta forma, incrementar el nivel de productividad de los grupos desarrolladores y minimizar los riesgos globales sin incurrir en gastos exorbitantes. Otra ventaja es poder integrar lo mejor de las tecnologías para desarrollar una aplicación de manera personalizada a la medida de las necesidades de los clientes. El DSBC, pertenece al paradigma de programación de sistemas abiertos, los cuales son extensibles y tienen una interacción con componentes heterogéneos que ingresan o abandonan el sistema de forma dinámica, es decir

que los componentes pueden ser remplazados, por otros independientemente de su arquitectura y desarrollo. (1)

1.1.1 Características claves para que un elemento sea catalogado como componente.

-Identificable: Debe tener una identificación que permita acceder fácilmente a sus servicios y que permita su clasificación.

– Auto contenido: Un componente no debe requerir de la utilización de otros para finalizar la función para la cual fue diseñado.

– Puede ser remplazado por otro componente, se puede remplazar por nuevas versiones u otro componente que lo reemplace y mejore.

– Con acceso solamente a través de su interfaz, debe asegurar que estas no cambiarán a lo largo de su implementación.

– Sus servicios no varían, las funcionalidades ofrecidas en su interfaz no deben variar, pero su implementación sí.

– Bien Documentado, un componente debe estar correctamente documentado para facilitar su búsqueda si se quiere actualizar, integrar con otros, adaptarlo, entre otros.

– Es genérico, sus servicios debe servir para varias aplicaciones.

– Reutilizado dinámicamente, puede ser cargado en tiempo de ejecución en una aplicación.

– Independiente de la plataforma, Hardware, Software, S.O. (1)

1.1.2 Tipos de componentes.

Existen varios tipos de componentes clasificados según criterios como su modificabilidad, granularidad, fabricante, tecnología usada; la clasificación de componentes es un proceso extenso, ya que un componente involucra en sí mismo varios atributos relevantes. En este segmento se mencionan una serie de clasificaciones de los mismos:

- Caja negra
- Caja blanca
- Componentes de uso específico
- Componentes de negocio

- Marcos (frameworks)
- Componentes de aplicación
- Componentes hechos en casa
- COTS- Component Off The Shelf
- Componentes imperativos (Módulos, funciones)
- Componentes OO (Clases)
- Componentes distribuidos (Componentes CORBA, Componentes.NET, Componentes J2EE, Servicios web)

Específicamente este trabajo de diploma tributará a los Componentes Orientados a Objetos y a los Servicios Web, ya que para medir la eficiencia es necesario que el procedimiento sea aplicado a un producto o componente terminado, listo para ser usado en un ambiente específico bajo condiciones específicas.

1.2 Calidad de un Producto Software

La calidad dentro de la Ingeniería de Sistemas no es más que la totalidad de aspectos y características de un producto o servicio que tiene que ver con su habilidad de satisfacer las necesidades declaradas o implícitas, la calidad de productos software es una preocupación a la que se dedican muchos esfuerzos. Sin embargo el software casi nunca es perfecto, se hace necesario seguir los siguientes aspectos para medir la calidad de este producto: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad, portabilidad, características propias del software que hay que controlar y asegurar, ya que es un producto inmaterial que no se fabrica, tampoco se degradan físicamente, sino que se desarrolla.

El software puede tener errores e incidencias pero no similares a los de cualquier equipo de carácter físico. La calidad del software se encuentra a la par con la calidad tradicional, pero un paso atrás, debido a que la calidad tradicional tiene varias décadas de historia, mientras que la calidad de software tiene 50 a 60 años, para lograrla es de vital importancia tanto la especificación como la evaluación del producto, con este fin la ISO (International Standards

Capítulo 1 Fundamentación Teórica

Organization) y la IEC (Internacional Electrotechnical Commission) desarrollaron el estándar ISO 9126/1991 que más tarde se dividió en dos estándares relacionados: ISO 9126, relacionado con la calidad del producto y el estándar 14598, relacionado con la evaluación. Estas normas logran medir la calidad del producto software a partir de métricas internas que no es más que la capacidad de un conjunto estático de atributos para satisfacer las necesidades declaradas e implícitas de un producto de software bajo ciertas condiciones especificadas; o métricas externas que es la capacidad de un producto software para desarrollar el comportamiento de un sistema de forma que satisfaga las necesidades declaradas e implícitas de un sistema utilizado bajo condiciones especificadas, la base sobre la cual las métricas son seleccionadas depende de los objetivos de negocio especificados para el producto y las necesidades del evaluador. (2)

Dentro de la serie ISO/IEC 9126 se incluyen los siguientes estándares:

- ISO/IEC 9126-1, presenta las propiedades que deberían tener las medidas a utilizar para comparaciones.
- ISO/IEC 9126-2, presenta las medidas de calidad externas.
- ISO/IEC 9126-3, referido a las medidas de calidad internas.
- ISO/IEC 9126-4, explica las medidas de calidad en el uso.

Dentro de la serie ISO/IEC 14598 se pueden destacar las siguientes normas:

- ISO/IEC 14598-1, enumera las medidas aplicables en la evaluación de software y los requisitos que aquellas deberían cumplir.
- ISO/IEC 14598-2, explica cómo realizar la planificación y gestión de la evaluación.
- ISO/IEC 14598-3, explica cómo elegir atributos que representen los requisitos de calidad.
- ISO/IEC 14598-4, regula el proceso y entre otras muchas cosas, proporciona ejemplos de métodos de evaluación, como listas, historiales o tablas y recomendaciones en el uso de los mismos.
- ISO/IEC 14598-5, especifica cómo adaptar el proceso general de evaluación a un entorno más específico.

- ISO/IEC 14598-6, trata sobre la documentación del proceso.

La calidad del producto de software se debe desglosar jerárquicamente en un modelo de calidad compuesto de características y sub-características que puede usarse como una lista de chequeo de problemas relacionados con la calidad. A continuación se dan las definiciones de cada una de estas características:

- Funcionalidad, es la capacidad del software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas cuándo el software se usa bajo las condiciones especificadas.
 - Confiabilidad, capacidad del producto de software para mantener un nivel de ejecución especificado cuando se usa bajo las condiciones especificadas.
 - Usabilidad, capacidad del producto de software de ser comprendido, aprendido, utilizado y de ser atractivo para el usuario, cuando se utilice bajo las condiciones especificadas.
 - Eficiencia, capacidad del producto de software para proporcionar una ejecución o desempeño apropiado.
 - Mantenibilidad, capacidad del producto de software de ser modificado. Las modificaciones pueden incluir las correcciones, mejoras o adaptaciones del software a cambios en el ambiente, así como en los requisitos y las especificaciones funcionales.
 - Portabilidad capacidad de producto de software de ser transferido de un ambiente a otro.
- (2)

1.3 Eficiencia: Estado del arte.

La eficiencia es la capacidad del producto software para proporcionar una ejecución o desempeño apropiado, en relación con la cantidad de recursos utilizados, usados bajo condiciones establecidas, entre los recursos se pueden incluir otros productos del software, la configuración del software y el hardware del sistema y los materiales, (por ejemplo el papel de la impresión o disquetes), para un sistema que se opera por un usuario, la combinación de

funcionalidad, confiabilidad, usabilidad y eficiencia puede ser medida externamente por la calidad en el uso. (2)

1.1.3 Otras definiciones de eficiencia.

Existen otros conceptos de eficiencia que ayudan a ampliar el empleado en esta investigación expuesto anteriormente, entre ellos se pueden mencionar los siguientes:

- Concepto óptimo de Pareto según el cual una asignación de recursos A es preferida a otra B, si y solo si con la segunda al menos algún individuo mejora y ninguno empeora, es decir un óptimo paretiano es una asignación de recursos que no puede modificarse para mejorar la situación de alguien sin empeorar la de otro(s). La garantía de la existencia de este tipo de equilibrio conlleva al cumplimiento de tres condiciones que están relacionadas al término de eficiencia: eficiencia productiva, de intercambio y global. La primera se cumple cuando existen iguales relaciones marginales, técnicas de sustitución entre los recursos empleados para generar outputs. La segunda cuando la relación marginal de sustitución entre los bienes son las mismas para todos los consumidores y la tercera necesita de la igualdad entre las relaciones marginales de sustitución entre pares de bienes y su relación marginal de transformación para la totalidad de los individuos. (3)
- Lindbeck(1971): consideró la diferenciación de tres extensiones adicionales de la idea de eficiencia: asignativa, técnica y coordinativa e informativa. La primera de ellas fuerza a que los inputs se agrupen en función del gusto de los individuos ocasionando de que curva de transformación se expanda; la segunda surge de la interpretación de la función de producción como el conjunto de los puntos fronteras del conjunto de producción quedando particionado así el espacio de asignaciones en eficientes, ineficientes e imposibles. Este se trata de un concepto puramente técnico puesto que contempla únicamente la relación entre las cantidades de insumos y sus productos no sus valores. En tercer lugar la eficiencia coordinativa e informativa se alcanza mediante la minimización de los costes de la información necesaria para la toma de decisiones. (3)
- Farrell (1957) fue el primero en dividir el concepto de eficiencia en dos componentes: técnica y asignativa, además de desarrollar un método para el cálculo empírico para

medir la eficiencia relativa. Este trabajo tuvo como antecedentes las investigaciones realizadas por Koopmans (1951) y Debreu (1951). Koopmans fue quien dio por primera vez una definición de eficiencia productiva; centrándose en la eficiencia técnica afirmó que una combinación factible de inputs y outputs es técnicamente eficiente, si es tecnológicamente imposible aumentar algún output o reducir algún input sin reducir simultáneamente al menos otro output o aumentar al menos otro input. Por otra parte Debreu (1951) propuso la construcción de un índice de eficiencia técnica, al que llamó “coeficiente de utilización de los recursos” que definía como la unidad menos la máxima reducción equiproporcional en todos los inputs para un nivel dado de outputs. (3)

1.1.4 Sub-Características.

Una de las características necesarias para medir la calidad del software es la eficiencia a la cual acompañan sus sub-características que ayudan a desglosar este concepto, haciéndolo de esta manera más aplicable:

- Rendimiento, capacidad del producto de software para proporcionar apropiados tiempos de respuesta y procesamiento, así como tasas de producción de resultados, al realizar su función bajo condiciones establecidas. (2)
- Utilización de recursos capacidad del producto de software para utilizar la cantidad y el tipo apropiado de recursos cuando el software realiza su función bajo las condiciones establecidas. (4)

1.1.5 Diferencias con la eficacia.

Los términos de eficacia y eficiencia suelen ser confundidos y en ocasiones hasta considerados sinónimos sin embargo en este contexto no significan lo mismo, pues eficacia es la capacidad del producto de software de permitir que los usuarios logren objetivos especificados con precisión e integridad en un contexto especificado y está más bien relacionado con la calidad durante el uso siendo esta la perspectiva que tiene el usuario de la calidad de un sistema que contiene el software, y es medida en lo que se refiere al resultado de usar el software, en lugar de las propiedades del propio software y puede ayudar a validar la calidad externa que no es más que el grado en la que un producto satisface necesidades explícitas e implícitas cuando se utiliza bajo condiciones especificadas. Mientras que la eficiencia empieza a jugar su papel desde

la propia construcción del software jugando un papel preponderante en la medición de la totalidad de atributos de un producto que determina su capacidad de satisfacer necesidades explícitas e implícitas cuando es usada bajo condiciones específicas. La noción de eficacia se refiere únicamente a la obtención de resultados, sin tener en cuenta los recursos empleados. La eficacia no implica, necesariamente, eficiencia, pero la eficiencia sí implica, como condición necesaria, pero no suficiente, a la eficacia; es decir, la eficiencia requiere de la obtención de resultados. La eficacia no implica efectividad, pues en condiciones normales se producen distintas influencias que pueden hacer que los resultados obtenidos varíen, sin embargo, la efectividad sí requiere eficacia. Lograr que los procesos sean eficientes, requieren de efectividad y eficacia. Los resultados obtenidos se pueden interpretar como cuantificación de la meta propuesta, aún en caso de no lograr los resultados esperados (5)

1.1.6 Herramientas para medir eficiencia.

Para todas las industrias hay marcos de trabajo compuestos por normativas y estándares más o menos estables y consensuados, que en forma de modelos ayudan a mejorar o evaluar la calidad de sus sistemas de producción. Según como afrontan las organizaciones el desarrollo del software, éste puede comportarse como factor de riesgo o amenaza para el negocio, o por el contrario como una poderosa oportunidad. Todas las empresas quieren producir más rápido, mejor y con menores costes, y sin duda esto es posible porque la naturaleza del software no es origen de riesgos y problemas, sino una fuente de oportunidades. Las empresas productoras de software han ido creando herramientas para medir la eficiencia en mayor o menor medida, entre las más completas podemos mencionar:

- ✓ Frontier Analyst, realiza la evaluación de la eficiencia mediante el análisis envolvente de datos. Es un software desarrollado y comercializado por la empresa Banxia Software Ltd. Actualmente se encuentra en el mercado la versión 3 que presenta entre otras las siguientes características básicas:
 - Capacidad para analizar desde 5 hasta un número “ilimitado” de Unidades (dependiendo de la licencia adquirida).
 - Hasta 32 variables Input y Output.

Capítulo 1 Fundamentación Teórica

- Las Unidades y variables Input/Output pueden ser habilitadas o deshabilitadas, de manera individual, en cada análisis.
- Editor de datos incorporado.
- Pegar datos directamente desde una hoja de cálculo.
- Importar datos desde ficheros delimitados por tabulaciones.
- Copiar tablas y gráficos al portapapeles para ser incluidas en informes.
- Copiar resultados al portapapeles para análisis posteriores (Nuevo en la versión 3).
- Modo de minimización Input y maximización Output.
- Modo de rendimientos constantes y variables a escala.
- Interfaz de fácil utilización por parte del usuario.
- Alta calidad en la visualización de gráficos de todos los aspectos del análisis.
- Gestión flexible de los datos disponibles al poder realizar filtrado de datos y habilitar o deshabilitar tanto Unidades como variables.
- Posibilidad de importar tanto desde un fichero como de una hoja de cálculo utilizando un “asistente”.
- La selección de variables Input/Output permite una rápida comparación entre Unidades.
- Impresión de informes de resultados.
- Posibilidad de establecer pesos al objeto de asegurar que los factores más importantes son siempre considerados.
- Informes de puntuaciones de eficiencia con diversos métodos de clasificación.
- Herramienta para diseñar informes personalizados (opcional).

Capítulo 1 Fundamentación Teórica

- Herramienta para publicar los resultados en formato web (Publish to Web) y facilitar así su distribución vía intranet/internet.
 - Eficiencia cruzada (opcional, para usuarios avanzados). (6)
- ✓ **Indicadores**, es una herramienta para medir la eficiencia en el uso de las tecnologías de la información y las comunicaciones en los centros de educación superior en Cuba. Con la integración de las TIC en los centros universitarios (intranet, pizarras digitales en las aulas, salas multiusuario, entre otras), se abren nuevas ventanas al mundo que permiten a estudiantes y profesores el acceso a cualquier información necesaria en cualquier momento, la comunicación con compañeros y colegas de todo el planeta para intercambiar ideas y materiales, para trabajar juntos. El Ministerio de Educación Superior (MES) ha trazado indicadores para garantizar la eficiente utilización de los medios informáticos. Dichos indicadores permitirán: Medir cambios a través del tiempo. Facilitan mirar de cerca los resultados de iniciativas o acciones. Utilizarlos como instrumentos para evaluar y dar surgimiento al proceso de desarrollo. “Indicadores” se propone que una organización, un proyecto, una nueva tarea, deben concretarse en expresiones medibles, que sirvan para expresar cuantitativamente dichos objetivos y tareas, y son los encargados de esa concreción. Esta herramienta adoptó como concepto de indicador los compendios, conjuntos o selecciones de datos básicos, que permite calificar un concepto abstracto y a su vez provea del detalle cuantitativo y cualitativo de un conjunto de objetivos con el fin de medir el progreso logrado con respecto a las metas. Partiendo que una de las utilidades de los indicadores es la toma de decisiones a continuación se enumeran diferentes tipos de indicadores que se pueden utilizar de acuerdo a las necesidades reales, que apoyaran la toma de acciones de acuerdo a sus resultados.
- **Indicadores Cuantitativos**, son los que se refieren directamente a medidas en números o cantidades.
 - **Indicadores Cualitativos**, son los que se refieren a cualidades. Se trata de aspectos que no son cuantificados directamente. Se trata de opiniones, percepciones o juicio de parte de la gente sobre algo.
 - **Indicadores Directos**, son aquellos que permiten una dirección directa del fenómeno.

Capítulo 1 Fundamentación Teórica

- Indicadores Indirectos, cuando no se puede medir de manera directa la condición económica, se recurre a indicadores sustitutos o conjuntos de indicadores relativos al fenómeno que nos interesa medir o sistematizar.
- Indicadores Positivos, son aquellos en los cuales si se incrementa su valor estarían indicando un avance hacia la equidad.
- Indicador Negativo, son aquellos en los cuales si su valor se incrementa estarían indicando un retroceso hacia la inequidad.

Después de definido que tipo de indicadores existen para verificar, o chequear la situación real los autores asumen que se deben tener en cuenta los aspectos relacionados de cómo realizar la medición pues con la misma se completará el proceso de la toma de decisiones.

La medición nos permite planificar con mayor certeza y confiabilidad. También discernir con mayor precisión las oportunidades de mejora de un proceso dado. Se puede analizar y explicar como han sucedido los hechos. (7)

- ✓ La medición de la eficiencia técnica mediante el Análisis Envolvente de Datos: en su forma operativa básica es una metodología utilizada para la medición de eficiencia comparativa de unidades homogéneas, es decir, que tienen una misma finalidad, la racionalidad. Partiendo de los insumos y productos, proporciona un ordenamiento de los agentes otorgándoles una puntuación de eficiencia relativa. De esta forma, los agentes que obtengan el mayor nivel de producto con la menor cantidad de insumos serán los más eficientes del grupo y por tanto, obtendrán los puntajes más altos. El método de estimación envolvente de datos evalúa la eficiencia de una unidad tomadora de decisiones refiriéndose al “mejor” productor. Considera que una unidad productiva es eficiente, y por tanto pertenece a la frontera de producción, cuando produce más de algún output sin generar menos del resto y sin consumir más inputs, o bien, cuando utilizando menos de algún input, y no más del resto, genere los mismos productos. La idea es comparar cada unidad no eficiente con aquella que lo sea, y a la vez tenga una técnica de producción similar, es decir, que utilice inputs en proporciones similares para producir outputs parecidos.

1.1.7 Las métricas para la medición de la característica eficiencia

Las métricas externas de eficiencia deben ser capaces de medir un atributo como es rendimiento o consumo de tiempo y el comportamiento en la utilización de los recursos del sistema, incluyendo al software, durante las pruebas y la implantación. Estas son:

- Métricas de rendimiento, miden el crono-comportamiento del sistema que no es más que el tiempo que demora el sistema en dar una respuesta determinada.
- Métricas de utilización de recursos, miden atributos de utilización de los recursos como los dispositivos de entrada y salida.

1.4 Definición de procedimiento.

Un procedimiento es la acción de proceder o el método de ejecutar algunas cosas. Se trata de una serie de pasos definidos que permiten realizar un trabajo de forma correcta, hace referencia a las normas de desarrollo del proceso, siendo una secuencia de pasos repetibles y determinista, esto quiere decir que para los mismos conjuntos de valores de salida, siempre se obtendrán los mismos conjuntos de valores de entrada.

Específicamente el tipo de procedimiento al que se refiere en esta investigación se trata de una secuencia de pasos finitos que van a dar solución al cómo medir lo que ya está descrito en las normas ISO/IEC 9126 e ISO/IEC 14598 sobre la eficiencia de software, compuesto esencialmente por términos y definiciones, normas generales, manuales del procedimiento, y el desarrollo del procedimiento.

Conclusiones Parciales.

En este capítulo se han visto los principales conceptos usados en la investigación llegando de esta manera a la conclusión que el Desarrollo de Software Basado en Componentes es una de las metodologías de desarrollo más importantes en la actualidad, la reducción en tiempos, costos y riesgos asociados en el proceso de construcción de software son sólo algunas de las razones que confirman su importancia, sin embargo se hace cada vez más necesario producirlos con calidad para de esta manera satisfacer las necesidades de los clientes y no ver comprometida la calidad del producto total, se profundiza en la característica de eficiencia

Capítulo 1 Fundamentación Teórica

debido a que es objetivo de la investigación, pues se considera de suma importancia el uso de los recursos y el tiempo de ejecución de los componentes para que éstos cumplan con sus expectativas. De igual manera se hace referencia a las normas de calidad que hoy guían este proceso y que especifican qué es lo que se debe hacer para obtener productos con calidad a partir de cada una de las características. Se especifica la diferencia entre eficiencia y eficacia ya que usualmente suelen ser confundidas. Por tanto se puede concluir en que se hace cada día más necesario obtener componentes software con el mayor grado de eficiencia posible garantizando de esta manera una parte de la calidad del producto software.

Capítulo 2 Solución propuesta

Introducción:

En este capítulo se define la solución del procedimiento que permitirá evaluar la eficiencia de componentes de software en el Centro de Desarrollo de Informática Industrial, está dividido en cuatro etapas: la primera de ellas es en la que se define la fase de desarrollo en la que se encuentra el componente según la metodología RUP, aunque también es aplicable a cualquier otra, esta descripción incluye las características que se necesitan conocer del componente para pasar a la segunda etapa donde se describe la evaluación de la eficiencia y cada una de las sub-características a través de las métricas internas y externas definidas en las normas ISO/IEC 9126 e ISO/IEC 14598, estableciendo luego rangos de valores que indican el grado de cumplimiento de la misma, luego se pasa a la parte práctica correspondiente a la tercera etapa donde se ejecutan las mediciones y se obtienen los resultados que serán procesados en la cuarta etapa de este procedimiento.

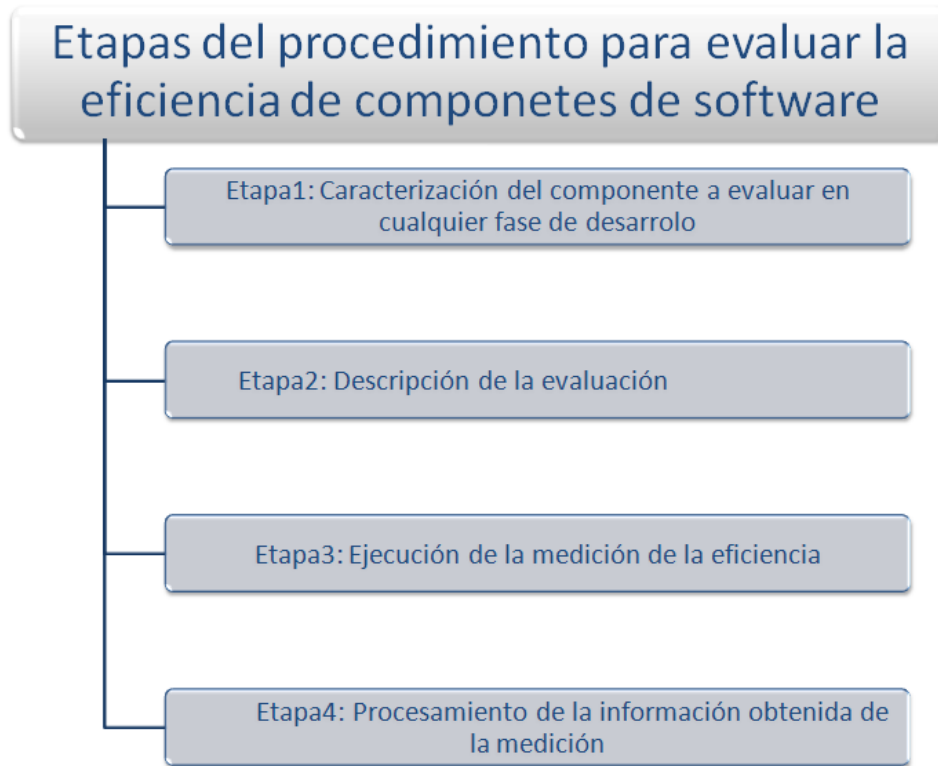


Ilustración 1: Etapas del procedimiento

2.1 Nombre del procedimiento

Evaluación de la eficiencia de componentes de software.

2.2 Objetivos

Proporcionar una serie de pasos finitos y repetibles que permitan evaluar la eficiencia de software basado en componentes en el Centro de Desarrollo de Informática Industrial (CEDIN).

2.3 Alcance

El procedimiento se podrá aplicar en todos los componentes que se desarrollen en el CEDIN.

2.4 Referencias.

Manual de procedimientos IPP-1000:2008.

ISO/IEC 14598 Evaluación de los productos de software

ISO/IEC 9126-3 Métricas Internas.

NC-ISO/IEC 9126-4. ANEXO B- Tablas.

2.5 Responsables.

Ejecuta: Administrador de Calidad y/o Desarrollador.

Responsable de su ejecución: Grupo de Gestión de la Calidad del CEDIN.

Revisa y actualiza este procedimiento: Jefe del grupo de Gestión de la calidad (GGC) del CEDIN.

Fiscaliza su cumplimiento: Jefe del grupo de Gestión de la calidad (GGC) del CEDIN.

2.6 Términos y definiciones.

RUP: Metodología de desarrollo de software cuyas siglas en inglés significan Rational Unified Process.

Rendimiento: Capacidad del producto de software para proporcionar apropiados tiempos de respuesta y procesamiento.

Componente de software: Paquete coherente de artefactos de software que puede ser desarrollado independientemente y entregado como una unidad.

2.7 Normas generales

El procedimiento, que está formado por una serie de pasos finitos, debe ser iterado secuencialmente a partir de las etapas antes descritas, para lograr el éxito en su ejecución es necesario que el equipo de trabajo integrado por los administradores de la calidad y los desarrolladores indistintamente, estén totalmente preparados para su aplicación, para ello es

necesario conocimiento sobre la eficiencia de componentes de software y la calidad de dichos componentes de manera general, lo que permitirá una mayor comprensión del procedimiento que deberán dominar a la perfección.

2.8 Etapas del procedimiento.

2.8.1 Etapa1: Caracterización del componente a evaluar en cualquier fase de desarrollo.

En esta etapa se describen una serie de actividades que permitirán un conocimiento lo más profundo posible del componente a evaluar, según la fase de desarrollo en la que se encuentra, que determinará en gran medida el rol que aplicará el procedimiento, ya que en dependencia de la misma el procedimiento podrá ser aplicado por el administrador de la calidad y/o el desarrollador, es importante también conocer el tipo de componente así como sus requerimientos. A continuación se muestran una serie de actividades que definen esta primera etapa.

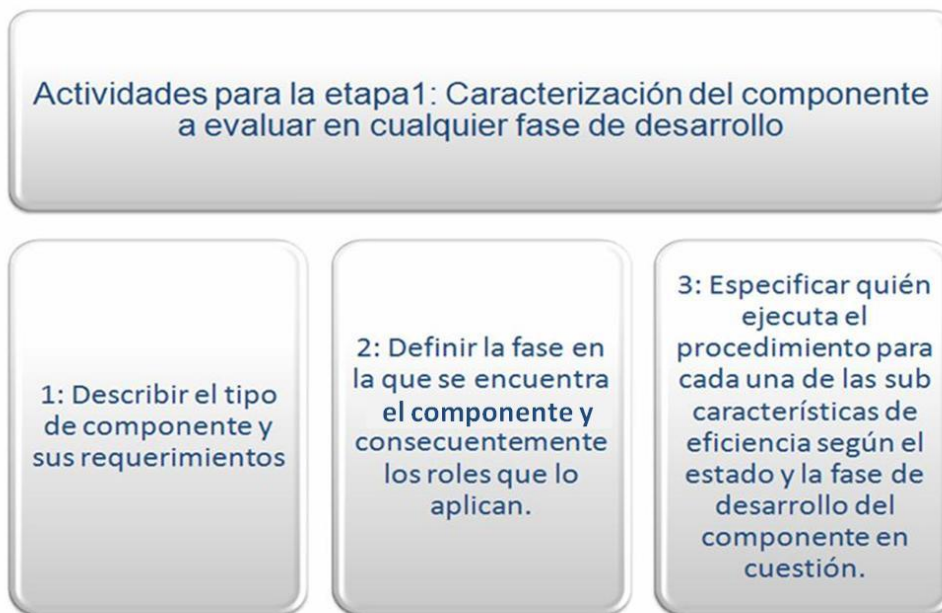


Ilustración 2: Actividades para la etapa 1

2.8.1.1 Descripción del tipo de componente y sus requerimientos.

A través de la consulta a diferentes artefactos del expediente del proyecto en dependencia de la fase en la que se encuentre el componente se hace una descripción de este, teniendo en cuenta fundamentalmente el tipo de componente, así como sus requerimientos, en la presente investigación se ha profundizado en dos tipos de componentes fundamentales, atendiendo a las necesidades del CEDIN:

- ✓ El primero de ellos los componentes distribuidos (Servicios web):

Los componentes de servicios web posibilitan que aplicaciones de diferentes plataformas tecnológicas puedan usar servicios de otras aplicaciones, los servicios web son pequeñas aplicaciones que pueden ser publicadas, localizadas e invocadas desde cualquier sitio web o red local basados en estándares abiertos de Internet. Combinan los mejores aspectos de la programación basada en componentes y la programación web, y son independientes del lenguaje en el que fueron implementados, del sistema operativo donde funcionan, y del modelo de componentes utilizado en su creación.

Los servicios web surgieron como aplicaciones remotas accesibles por Internet (usando Servlets, JSP, ASP y otra tecnología web) utilizadas para la programación web en entornos de comercio electrónico. Con la consolidación de XML, los servicios web empezaron a utilizar mensajes en XML para la consecución de transacciones distribuidas utilizando el protocolo de transferencia HTTP. Hoy día los repositorios de servicios web son un punto de referencia muy importante para los desarrolladores de aplicaciones basadas en componentes web.

- ✓ El segundo Componentes Orientado a Objeto (Clases):

El desarrollo de estos componentes tiene gran importancia pues posibilita en gran medida la reutilización de código y la extensión de este en sistemas complejos, agilizando de esta manera su desarrollo y mantenimiento. Estos sistemas están basados en el paradigma de la programación orientada a objetos, donde dichos objetos son tratados como bloques de construcción lógicos, siendo cada objeto una instancia de una clase y relacionándose las clases entre si, logrando de esta manera relacionar al sistema con el mundo real y facilitando el trabajo en equipo.

- La eficiencia tiene dos sub características fundamentales: la primera de ellas es el rendimiento, capacidad del producto de software para proporcionar apropiados tiempos de respuesta y procesamiento, así como tasas de producción de resultados, al realizar su función bajo **condiciones establecidas** y la segunda es la utilización de los recursos, capacidad del producto de software para utilizar la cantidad y el tipo apropiado de recursos cuando el software realiza su función bajo las **condiciones establecidas** que dependen en gran medida de los requerimientos del componente, es por ello que se hace necesario conocer:
- La versión del sistema operativo
 - El IDE de programación utilizado
 - La capacidad de la memoria RAM
 - Otras necesidades de hardware como: El uso de dispositivos externos, el tipo de microprocesador, la capacidad del disco duro y otras.

2.8.1.2 Definir la fase en la que se encuentra el componente y consecuentemente los roles que lo aplican.

Las fases que se definirán a continuación son las pertenecientes a la metodología RUP, que es la que se está utilizando actualmente por la mayoría de los proyectos del centro, sin embargo el procedimiento puede ser aplicado por un proyecto desarrollado bajo cualquier otra metodología, basta solo con especificar el nombre de la misma y la fase en la que se encuentra, para poder de esta manera asignar el/los rol/es que aplicarán el procedimiento.

La metodología RUP se divide en cuatro fases:

- **Inicio:** El Objetivo en esta etapa es determinar la visión del proyecto.
- **Elaboración:** Se especifica en detalle la mayoría de los casos de usos del producto y se diseña la arquitectura del sistema. El resultado de esta fase es una línea base de la arquitectura.

- **Construcción:** En esta fase se crea el producto, la línea base de la arquitectura crece hasta convertirse en el sistema completo.
- **Transición:** El producto se convierte en versión beta, un número reducido de usuarios con experiencia prueban el producto e informa de defectos y deficiencias. Los desarrolladores corrigen los problemas e incorporan algunas de las mejoras sugeridas.

Los roles que podrán aplicar el procedimiento son:

- El administrador de la calidad: quien participa en el análisis y recolección de datos para la elaboración de las métricas, elabora el plan de mediciones para medir el rendimiento y la utilización de los recursos, colabora en las auditorías realizadas al proyecto, guía el diseño y ejecución de las pruebas que comprobarán el grado de eficiencia del software, por tanto debe ser una persona con gran conocimiento sobre la calidad de los componentes de software y específicamente sobre la eficiencia y sus sub-características, comprometido con el centro y con los proyectos en los que trabaja, para lograr realizar evaluaciones que tributen a un mejor desarrollo del software basado en componentes.
- El desarrollador: debe aplicar eficientemente sus amplios conocimientos sobre técnicas de programación y diseño de algoritmos, poner todas sus habilidades lógicas en crear software cada vez más eficientes, entre sus responsabilidades se encuentran: diseñar la arquitectura del software de manera tal que sea lo más eficiente posible, convirtiendo la especificación del sistema en código fuente ejecutable, integrar los componentes del sistema sin perder el grado de eficiencia de que logra cada uno por separado, analizar, procesar y evaluar los resultados de la aplicación del procedimiento.

2.8.1.3 Especificar quién ejecuta el procedimiento para cada una de las sub-características, según el estado y la fase de desarrollo del componente.

El procedimiento puede ser ejecutado, por dos roles, el desarrollador o el administrador de la calidad, que lo harán en dependencia del estado del componente (desarrollo o terminado): en desarrollo, a esta etapa puede corresponder cualquier fase del RUP, pues comprende desde que se determinan los objetivos y la visión del proyecto hasta que este es liberado, y es

precisamente en este momento donde el componente pasa al estado de terminado que correspondería en su primera versión a la fase de transición del RUP.

La eficiencia tiene dos sub-características: El rendimiento y la utilización de los recursos, las que deben ser evaluadas tanto en el estado de desarrollo como en el de terminado, a continuación se muestra una tabla donde se resume lo antes expuesto:

Estado	Fase	Roles		Sub-características	
		Administrador de la calidad	Desarrollador	Rendimiento	Utilización de recursos
Desarrollo	Inicio		x	x	x
	Elaboración		x	x	x
	Construcción	x	x	x	x
	Transición	x	x	x	x
Terminado	Inicio				
	Elaboración				
	Construcción				
	Transición	x	x	x	x

Tabla 1: Correspondencia de las sub-características por estado fase y rol

2.8.2 Etapa2: Descripción de la evaluación.

En esta etapa se definen teóricamente dos aspectos fundamentales, la identificación de las métricas internas y externas, que se aplicarán a cada sub-característica de la eficiencia y luego se establecerán criterios de evaluación a través de los cuales se pueden analizar los resultados obtenidos al aplicarlas, lo que permitirá arribar a conclusiones sobre el estado de la eficiencia del componente a evaluar.

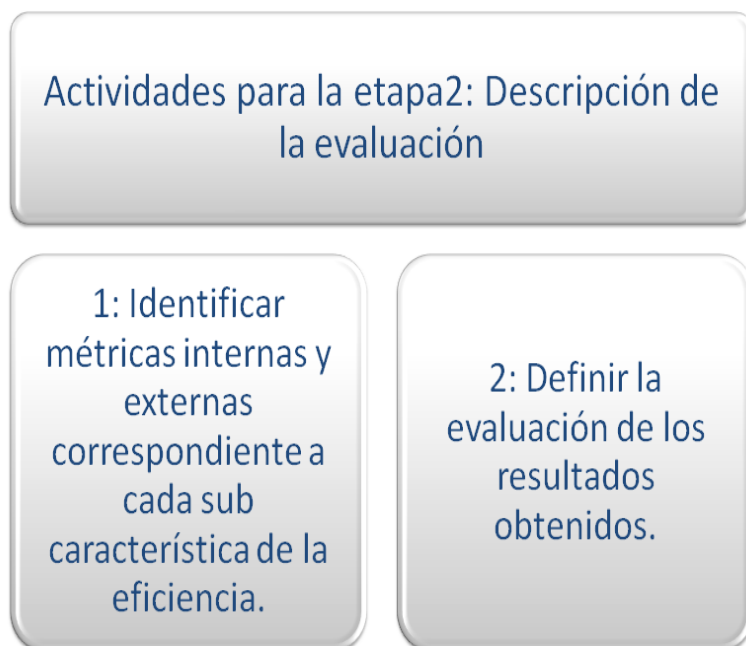


Ilustración 3: Actividades para la etapa 2

2.8.2.1 Identificar las métricas internas y externas correspondientes a cada sub-característica de la eficiencia.

Para evaluar satisfactoriamente la eficiencia de cualquier componente software es necesario aplicar métricas a cada una de las sub-características con las que se podrá obtener un valor cuantitativo de este requisito tan importante para lograr la calidad de los productos, y a partir de estos resultados aplicar acciones correctivas que permitan mejorarlo total o gradualmente, o en última instancia al menos conocer el grado de dificultad existente en este aspecto. A continuación se definen las métricas a utilizar por sub-característica:

Sub Característica	Nombre de la métrica	Definición	Metas	Procedimiento de análisis	Ejecutores
---------------------------	-----------------------------	-------------------	--------------	----------------------------------	-------------------

Capítulo 2 Solución Propuesta

Rendimiento	Tiempo de respuesta	T = Tiempo empleado en obtener el resultado	¿Cuánto tiempo toma completar una tarea en particular? ¿Cuánto toma antes de que el sistema responda a determinada operación?	Comience una tarea específica. Mida el tiempo que ésta toma en completar una muestra hasta que concluya la operación emprendida. Guarde los registros de cada intento.	Desarrollador Administrador de la calidad
--------------------	---------------------	---	--	--	--

Capítulo 2 Solución Propuesta

	<p>Tiempo de espera</p>	<p>$X = T_a/T_b$</p> <p>T_a - Tiempo total dedicado a esperar</p> <p>T_b - Tiempo consumido por la tarea</p>	<p>¿Qué proporción del tiempo los usuarios dedican a esperar por una respuesta del sistema?</p>	<p>Ejecutar los casos de pruebas en situaciones características y tareas concurrentes. Mida el tiempo que toma completar las operaciones seleccionadas. Guarde un registro de cada intento y calcule el tiempo medio para cada caso o situación.</p>	<p>Desarrollador</p> <p>Administrador de la calidad</p>
<p>Utilización de recursos</p>	<p>Tiempo de espera del usuario ante la utilización de los equipos de E/S</p>	<p>T = Tiempo empleado en esperar por la culminación de una operación de un equipo de E/S</p>	<p>¿Cuál es el impacto de la utilización de los equipos de E/S en el tiempo de espera de los usuarios?</p>	<p>Ejecute un amplio número de tareas concurrentes y mida el tiempo de espera del usuario ante la utilización de los equipos de E/S</p>	<p>Desarrollador</p> <p>Administrador de la calidad</p>

	Relación entre los errores de memoria y el tiempo	$X = A/T$ $A =$ Número de mensajes de alerta o fallos del sistema $T =$ Tiempo de operación del usuario	¿Cuántos errores de memoria fueron identificados durante el período de tiempo especificado, y con una utilización de recursos especificada?	Calibre las condiciones de prueba. Emule una condición en que el sistema haya alcanzado un máximo de carga. Ejecute la aplicación y registre el número de errores debido a fallos de memoria	Desarrollador Administrador de la calidad
--	---	---	---	--	---

Tabla 2. Métricas por sub-características

2.8.2.2 Definir la evaluación de los resultados obtenidos.

En este acápite se define cómo evaluar y aplicar las métricas de acuerdo a la tabla anteriormente expuesta. A continuación se explicará el significado cuantitativo de las variables por cada sub-característica:

❖ Rendimiento:

➤ Tiempo de respuesta:

- $0 < T$ A mayor prontitud (menor tiempo) resultará más eficiente el rendimiento de la aplicación.
- Escala: La escala de tiempo es relativa a la unidad de medida con la que se mida el tiempo. $T =$ Tiempo
- Fuentes: Informe de evaluación Reporte de operación, midiendo el tiempo.

➤ Tiempo de espera:

- $0 \leq X$ Cuanto menor resultará mejor.
- Escala: Absoluta Tipo de medida: $X = \text{tiem}/\text{tiem}$ $T_A = \text{tiempo}$ $T_B = \text{tiempo}$.
- Fuentes: Informe de evaluación Reporte de operación, midiendo el tiempo.

❖ Utilización de los recursos:

➤ Tiempo de espera del usuario ante la utilización de los equipos de E/S:

- $0 < T$ Más corto resultará mejor.
- Escala: Relativa Tipo de medida: $T = \text{Tiempo}$.
- Fuentes: Informe de evaluación Reporte de operación, midiendo el tiempo.

➤ Relación entre los errores de memoria y el tiempo:

- $0 < X$ Más pequeña, mejor.
- Escala: Relativa Tipo de medida: $x = A/T$ $A = \text{contable}$ $T = \text{tiempo}$.
- Fuentes: Reporte de operación, midiendo el tiempo.

Sin embargo, como se puede apreciar existe más de una métrica para evaluar cada sub-característica por lo que se hace necesario, unificar las mismas en una sola fórmula que mostraremos a continuación.

➤ Rendimiento:

$$R = (T_R + T_E)/2$$

$R = \text{rendimiento}$, $T_R = \text{tiempo de respuesta}$, $T_E = \text{tiempo de espera}$

➤ Utilización de los recursos:

$$U_r = (T_{eu} + R_e)/2$$

Ur= Utilización de los recursos, Teu = Tiempo de espera del usuario ante la utilización de los equipos de E/S, Re= Relación entre los errores de memoria y el tiempo.

Los niveles de eficiencia de cada una de las sub-características estarán dados en la siguiente escala:

Nivel de eficiencia	Intervalo
Alto	$0 \leq X < 1$
Medio	$1 \leq X < 1.5$
Bajo	$X \geq 1.5$

Tabla 3: Niveles de eficiencia

Donde X es el valor de cada sub-característica.

2.8.3 Etapa3: Ejecución de la medición de la eficiencia.

Esta es la etapa del procedimiento donde realizan los cálculos explicados en la etapa anterior y se tabulan los resultados para un análisis posterior. En esta etapa se debe tener extremo cuidado al realizar los cálculos y medir los tiempos pues, un error en ello conduciría indudablemente al fracaso de la evaluación de la eficiencia de cualquier componente.

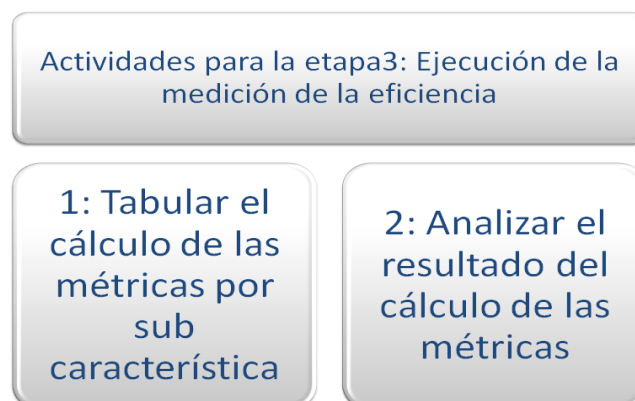


Ilustración 4: Actividades de la etapa3

2.8.3.1 Tabular el cálculo de las métricas por sub-característica:

En esta actividad se calculan las métricas y se guardan los resultados en la siguiente tabla para luego ser analizados.

Sub-característica	Métrica	Resultado
Rendimiento	Tiempo de respuesta	
	Tiempo de espera	
Utilización de recursos	Tiempo de espera del usuario ante la utilización de los equipos de E/S	
	Relación entre los errores de memoria y el tiempo	

Tabla 4: Resultado de las métricas

2.8.3.2 Analizar el resultado del cálculo de las métricas.

Esta actividad es totalmente dependiente de la anterior pues es a partir de ella de donde se podrá analizar y calcular los valores de las sub-características y ubicarlos en el nivel según la escala anteriormente definida, para luego poder tomar acciones correctivas de los mismos en caso que sea necesario. Estos valores se mostrarán en la siguiente tabla:

Sub-característica	Valor	Nivel
Rendimiento		
Utilización de recursos		

Tabla 5: Resultado de las sub-características

El valor de la sub-característica se obtiene luego de haber aplicado la fórmula definida en la actividad 2.8.2.2, y el nivel se obtiene comparando los resultados obtenidos con la escala definida en la misma actividad.

2.8.4 Etapa4: Análisis de la información obtenida de la medición.

En esta última etapa es donde se define como será el procesamiento de toda la información obtenida anteriormente, esta etapa es de suma importancia pues es donde se deja la constancia de la evaluación hecha, se genera el artefacto de salida del procedimiento que constituye un documento importante para posteriores análisis de dicha evaluación y va incluido en el plan de aseguramiento de la calidad (PPQA) del expediente del proyecto al cual pertenece el componente a evaluar.

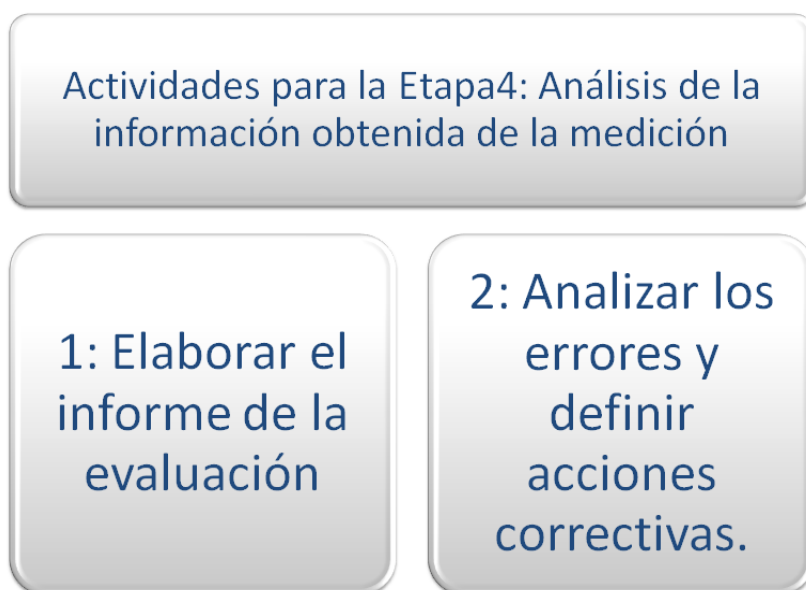


Ilustración 5: Actividades para la etapa4

2.8.4.1 Elaborar el informe de la evaluación.

Este informe es el artefacto que genera la aplicación del procedimiento, su estructura está expuesta en el anexo 1, donde quedan recogidos los resultados del cálculo de las métricas correspondientes a las sub-características rendimiento en la que se calcula el tiempo de espera y el tiempo de respuesta y la utilización de los recursos calculándose tiempo de espera del usuario ante la utilización de los equipos de E/S y la relación entre los errores memoria tiempo.

Al haber más de una métrica por sub-característica el informe recoge también el resultado de la aplicación de las fórmulas que aúnan las métricas en un solo resultado, de igual manera y no

menos importante es el análisis de la escala de comparación de niveles de las sub-características, dando una evaluación cuantitativa de cada una de ellas.

2.8.4.2 Analizar los errores y definir acciones correctivas.

Al aplicar el procedimiento pueden surgir varios errores que comprometan la eficiencia del componente, por lo que se hace necesario, tomar acciones correctivas. Para ello el informe también consta de la siguiente tabla que se debe llenar una vez concluida la ejecución del procedimiento y que será utilizada en las próximas iteraciones de este.

Errores encontrados	Acciones correctivas

Tabla 6: Errores y acciones correctivas

A continuación se muestran algunos de los errores más comunes y sus acciones correctivas:

Errores comunes	Acciones correctivas
Tiempo empleado en obtener el resultado demasiado largo(en la escala mayor que 1.5)	Verificar la compatibilidad de hardware o analizar en el caso del desarrollador cuán óptimo es el código.
Tiempo empleado en esperar por la culminación de una operación de un equipo de E/S demasiado largo(en la escala mayor que 1.5)	Verificar la compatibilidad con el sistema operativo, comprobar la integridad del dispositivo
Gran cantidad de errores de memoria tiempo (más de 5 para cada ejecución)	Ponerse en contacto con el equipo de desarrollo para una verificación de código
Uso excesivo de recursos	Ponerse en contacto con el equipo de desarrollo para una optimización de código

Tabla 7: Errores comunes y sus correspondientes acciones correctivas

Conclusiones Parciales

En este capítulo se expone la propuesta de procedimiento para evaluar la eficiencia de los componentes de software del CEDIN, a partir de las cuatro etapas propuestas se podrá sin lugar a dudas, profundizar en las características del componente a evaluar, aplicar la métricas que evalúan la eficiencia, calcular resultado de dicha evaluación, ubicar los niveles de eficiencia según la escala aquí definida, y por último procesar los datos y archivarlos en el Informe del procedimiento de la investigación que constituye el artefacto de salida de la misma y puede ser utilizado para próximas iteraciones del procedimiento.

Capítulo 3 Validación del procedimiento

Introducción:

En este capítulo se realiza la validación del procedimiento para evaluar la eficiencia de los componentes software del CEDIN, específicamente se validaron los componentes HMI y Seguridad del proyecto Energía UCI.

3.1 Descripción del procedimiento.

En este acápite se realiza la descripción del procedimiento común a ambos componentes pues los dos pertenecen al proyecto Energía UCI del SCADA-ux.

- El propósito de la aplicación de este procedimiento es conocer el grado de eficiencia de los componentes antes mencionados, en cuanto a las sub-características rendimiento y utilización de los recursos, así como la detecciones de posibles errores en los mismos y la aplicación de acciones correctivas en caso de encontrar alguno.
- El rol que ejecuta el procedimiento es el Administrador de la calidad
- La computadora en la que se ejecutó el procedimiento tenía los siguientes requerimientos de hardware: Procesador Intel(R) Pentium(R) Dual- Core CPU 3.00GHz, memoria instalada (RAM)1 GB, disco duro 250 GB y motherboard Dell.

Es necesario aclarar que el procedimiento es válido siempre que al componente a evaluar sea posible aplicarle al menos una de las métricas, en este caso al componente HMI no se le aplicó la métrica, Tiempo de espera del usuario ante la utilización de los equipos de E/S, pues el componente no utiliza dispositivos de entrada o salida, esta métrica es una de las dos utilizadas para medir la sub-característica utilización de los recursos, en este caso el valor de la misma estaría dado por el resultado de la aplicación de la otra métrica que la mide; al componente Seguridad solo fue posible aplicarle la métrica Tiempo de respuesta pues este carece de interfaz visual, necesaria para aplicar el resto de las métricas, por lo que la sub-característica utilización de los recursos no será medida en este componente y su nivel de eficiencia lo determinará el

Capítulo 3 Validación del procedimiento

rendimiento mostrado al aplicar la métrica anteriormente dicha. Lo anteriormente argumentado se resume en la siguiente tabla:

Sub-características	Métricas	HMI	Seguridad
Rendimiento	Tiempo de respuesta	X	X
	Tiempo de espera	X	-
Utilización de los recursos	Tiempo de espera del usuario ante la utilización de los equipos de E/S	-	-
	Relación entre los errores de memoria y el tiempo	X	-

Tabla 8: Métricas que se aplican por sub-características

3.1.1 Componente HMI.

3.1.1.1 Descripción del tipo de componente:

El nombre del componente es HMI (Human Machine Interface) son sus siglas en inglés, se encuentra clasificado entre los componentes imperativos (Módulos), utilizando para su desarrollo la programación Orientada a Objetos, se encuentra en el estado de terminado, el proyecto al pertenece este componente está inmerso en el programa de mejora ubicando al mismo en la fase de despliegue que este propone.

HMI tiene dos vistas una para Ejecución y otra para la Edición, la eficiencia de su ejecución depende en gran medida del tráfico y la velocidad de la red en ese momento, pues este permite que desde cualquier PC se puedan acceder a los datos del servidor logrando la supervisión y el control de forma gráfica de los procesos y el sistema.

Capítulo 3 Validación del procedimiento

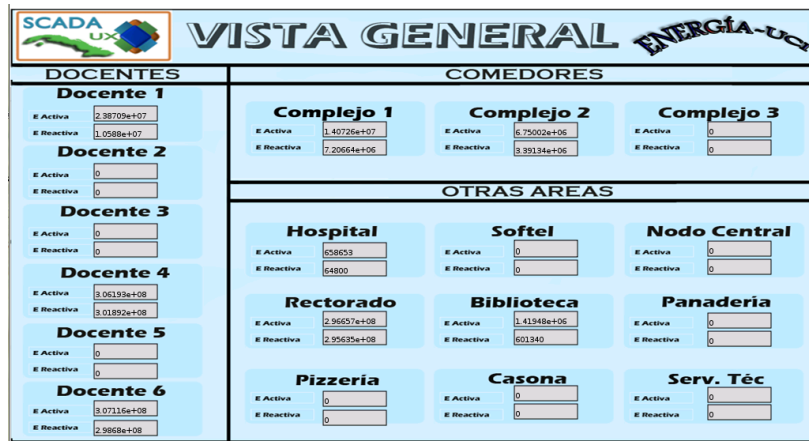


Ilustración 6: Vista ejecución de HMI

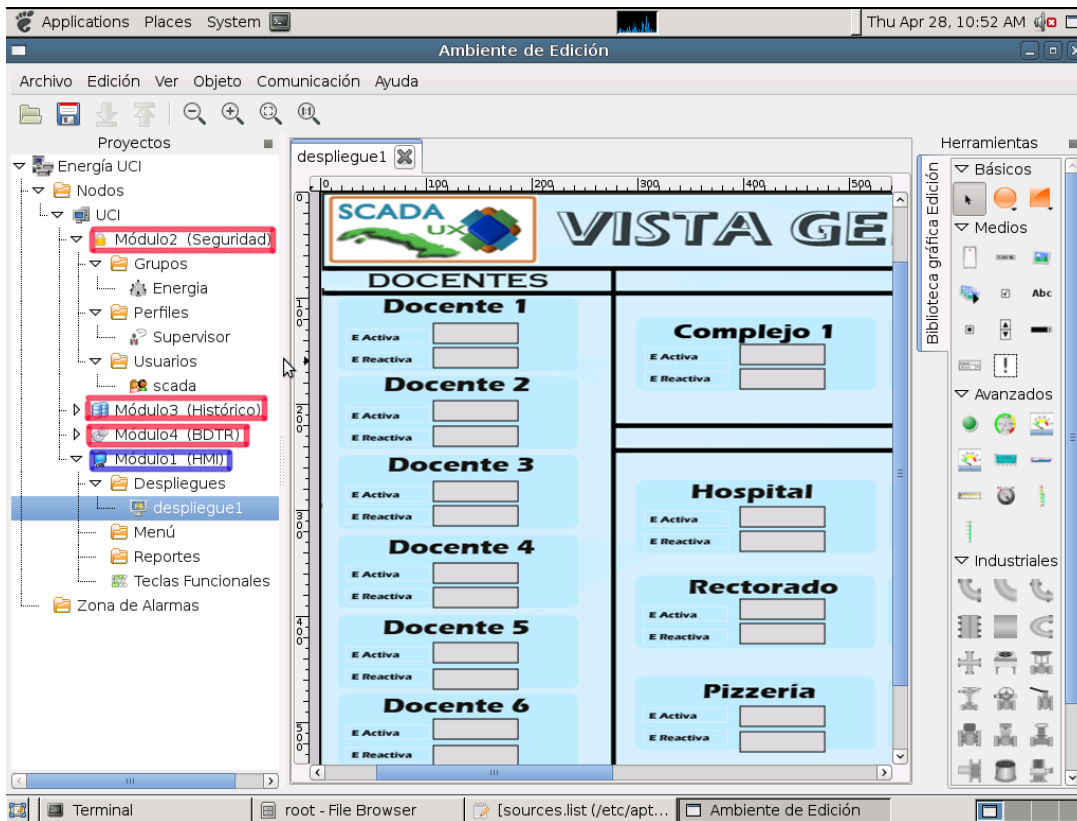


Ilustración 7: Vista edición de HMI

3.1.1.2 Cálculo de las métricas:

❖ Rendimiento:

- Al aplicar la métrica Tiempo de respuesta se calcula que el tiempo empleado en obtener el resultado es:

$$T = 1 \text{ s.}$$

Siendo un buen resultado a pesar de haber sido aplicada en un horario donde el tráfico en la red es muy elevado pudiendo incrementar el tiempo real de respuesta.

- El resultado de la aplicación de la métrica Tiempo de espera, que no es más que la proporción del tiempo los usuarios dedican a esperar por una respuesta del sistema, fue el siguiente:

$$X = 5.3/5 = 1.06$$

El promedio de estas dos métricas da un valor de 1.03 s siendo este el valor final de la sub-característica de rendimiento.

❖ Utilización de los recursos:

En esta sub-característica como solo se aplica la métrica Relación entre los errores de memoria y el tiempo, su valor coincide con el de la métrica.

$$X = 0/1 = 0$$

El sistema no presentó ningún error de memoria durante la ejecución del procedimiento por tanto sus resultados fueron óptimos.

A continuación se muestran los resultados por métricas:

Sub-característica	Métrica	Resultado
Rendimiento	Tiempo de respuesta	1
	Tiempo de espera	1.06

Capítulo 3 Validación del procedimiento

Utilización de recursos	Tiempo de espera del usuario ante la utilización de los equipos de E/S	-
	Relación entre los errores de memoria y el tiempo	0

Tabla 9: Resultado de la aplicación del procedimiento a HMI

En la siguiente tabla se muestran los resultados por sub-características:

Sub-característica	Valor	Nivel	Error
Rendimiento	1.03	Alto	+ - 0.5
Utilización de recursos	0	Alto	0

Tabla 10: Niveles por sub-características para HMI

En el caso de la sub-característica Rendimiento existe un error de +- 0.5, debido al tiempo de latencia que se produce por el abarrotamiento de la red, pues para ejecutar ese componente es necesario conectarse a un servidor. Es valido aclarar que la validación se hizo en un horario de un gran tráfico en la red, por lo que el error es el mayor posible y teniendo en cuenta su valor el nivel de esta sub-característica es alto.

3.1.1.3 Análisis de resultados:

Como se puede observar al aplicar las métricas al componente HMI perteneciente al proyecto Energía UCI del SCADA-ux, todas arrojan resultados positivos, no encontrando errores en el componente que comprometa la eficiencia y concluyendo que este componente es totalmente eficiente teniendo altos niveles de rendimiento y utilización de los recursos.

3.1.2 Componente Seguridad

3.1.2.1 Descripción del tipo de componente:

El nombre del componente es Seguridad, se encuentra inmerso en el programa de mejora, dentro de la fase de despliegue que este propone, esta clasificado entre los componentes imperativos (Módulos), utilizando para su desarrollo la programación Orientada a Objetos y se encuentra en el estado de terminado.

Capítulo 3 Validación del procedimiento

Este componente carece de interfaz visual pues su objetivo es proveer las funcionalidades necesarias para garantizar el trabajo a los usuarios autorizados, garantizando la protección contra ataques maliciosos o incluso involuntarios por parte tanto de personas como de recursos, entre los que se pueden encontrar fallos de electricidad o problemas de red.

3.1.2.2 Cálculo de las métricas.

En este componente como se explicó anteriormente solo se aplica la métrica Tiempo de respuesta correspondiente a la sub-característica de rendimiento, cuyo cálculo dio un valor de:

$$T=0.01s$$

El valor del rendimiento de este componente es excelente pues su tiempo de respuesta es instantáneo.

A continuación se muestran los resultados:

Sub-característica	Valor	Nivel
Rendimiento	0.01	Alto

Tabla 11: Niveles de rendimiento para Seguridad

3.1.2.3 Análisis de resultados:

Los resultados de la aplicación del procedimiento en el componente seguridad son impresionantes, su valor es muy próximo a 0 por lo que alcanza un nivel de rendimiento alto, no observándose errores al ejecutarlo.

3.2 Comparación entre los componentes:

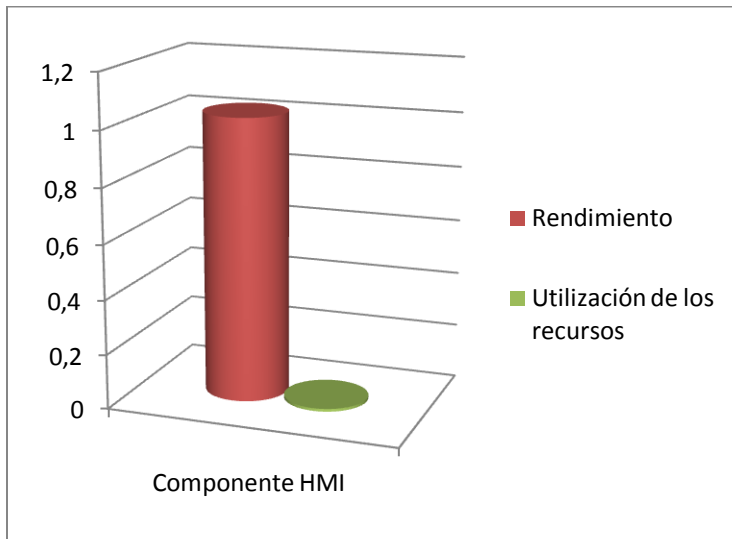


Ilustración 8: Resultado del procedimiento para HMI

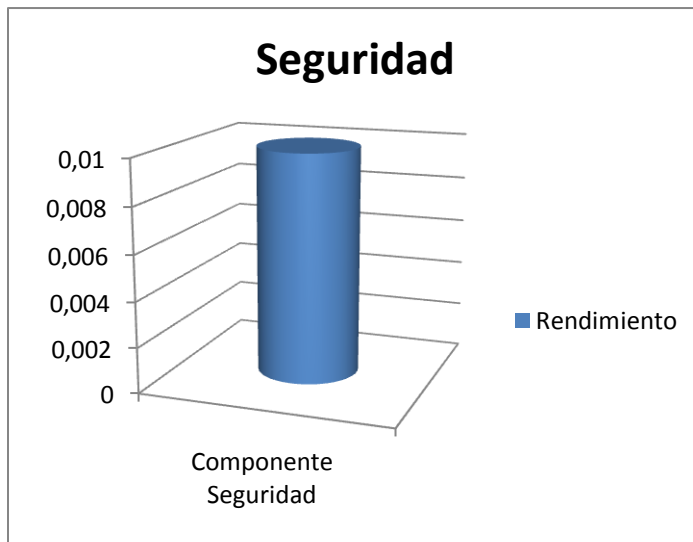


Ilustración 9: Resultado del procedimiento para Seguridad

Capítulo 3 Validación del procedimiento

Ambos componentes alcanzaron resultados elevados, la diferencia que se observa entre el rendimiento y la utilización de los recursos de HIM está dada porque en el componente no se utilizan dispositivos de entrada o salida y el mayor peso de la medición de la eficiencia cae en el rendimiento sin embargo aún así los resultados de esta sub-características son muy positivos, en el caso de seguridad a pesar de no medírsele la utilización de los recursos por la carencia de interfaz visual, los resultados del rendimiento son magníficos, por lo que se concluye que ambos componentes poseen características de eficiencia similares, pues corresponden al mismo proyecto y se encuentran en estado terminado, integrados y listos para usarse eficientemente.

Conclusiones Parciales:

En este capítulo se aplicó el procedimiento a los componentes HMI y Seguridad del proyecto Energía UCI, llegando a la conclusión de que ambos son totalmente eficientes, pues los resultados alcanzados los ubican en un nivel elevado de cada una de las sub-características que miden la eficiencia.

CONCLUSIONES GENERALES:

Al realizar este trabajo de diploma se le ha dado cumplimiento a su objetivo general, obteniendo un procedimiento que permite evaluar la eficiencia de los componentes software desarrollados en CEDIN.

Aportando a los Administradores de calidad y Desarrolladores, una serie de pasos consecutivos y finitos que explican cómo llevar a cabo lo que está definido en las normas ISO/IEC 9126 e ISO/IEC 14598.

Además este procedimiento se aplicó a dos componentes del CEDIN, pertenecientes al proyecto Energía UCI, obteniendo resultados satisfactorios, lo que valida su realización.

RECOMENDACIONES

Para dar continuidad a este trabajo de diploma se recomienda:

- Ejecutar el procedimiento propuesto en todos los componentes que se desarrollen en el CEDIN y de esta manera contribuir a una mayor calidad de los productos liberados.
- Elaborar una herramienta donde se puedan gestionar los resultados de la aplicación del procedimiento.
- Capacitar a los administradores de la calidad y desarrolladores del CEDIN para la correcta aplicación del procedimiento.

REFERENCIAS BIBLIOGRÁFICAS:

- (1) **Rojas, Maribel Ariza.** *Introducción y principios básicos del desarrollo de software basado en componentes.* 2004
- (2) **ISO, 9126.** “*Software product evaluation – Quality characteristics and guidelines for their use*”. 2005.
- (3) **14598, ISO. 1999.** *Evaluación de los productos software.* 1999.
- (4) **Gómez García, Oscar.** *SQuaRE: Una unificación de normas para la especificación de requisitos y la evaluación de la calidad.* Enero de 2009.
- (5) **Biblioteca Virtual.** *Eficiencia de la gestión de institutos públicos.* 2009
- (6) **Coll Serrano, Vicente.** *Fontier Analys, una herramienta para medir eficiencia.* 2006
- (7) **Pin, Enrique.** *Indicadores, una herramienta para medir la eficiencia en el uso de las tecnologías de la información y las comunicaciones en los centros de educación superior en Cuba.* 2006.

BIBLIOGRAFÍA:

14598, ISO. 1999.*Evaluación de los productos software.* 1999.

ISO, 9126.“*Software product evaluation – Quality characteristics and guidelines for their use*”. 2005.

Rojas, Maribel Ariza y García, Juan Carlos Molina. 2004.*INTRODUCCIÓN Y PRINCIPIOS BÁSICOS DEL DESARROLLO DE SOFTWARE BASADO EN COMPONENTES.* 2004.

0516_Roles y responsabilidades. *Calisoft UCI, 2009.*

IPP-1000:2008 *Elaboración y aprobación de los procedimientos y lineamientos para la actividad productiva.* *Calisoft UCI, 2008*

Garcia, Daysi Karina. *Gestión de la Calidad, Guía Didáctica.* Loja: s.n., 2008.

Gómez García, Oscar. *SQuaRE: Una unificación de normas para la especificación de requisitos y la evaluación de la calidad.* Enero de 2009.

Menéndez-Barzanallana Asensio, Rafael. *Metodología de desarrollo de software.* 2007.

Cueva Lovelle, Juan Manuel. *Calidad del Software.* Madrid, España. 21 de Octubre de 2003.

ANEXOS.

Anexo 1: Informe de Evaluación de la Eficiencia

	<p>Informe</p> <p>Evaluación de la Eficiencia de Componentes Software</p>
---	---

1 Descripción del componente que se evaluará

<Breve descripción del componente que se evaluará, así como de sus requerimientos>

- Estado y Fase del componente a evaluar.

<Se especifica estado y fase del componente que se evaluará auxiliándose del expediente del proyecto>

- Identificar el rol que ejecuta la evaluación

<Se especifica el rol que ejecuta el procedimiento>

- Identificar quién ejecuta el procedimiento según la fase y el rol

<Se identifican las sub-características de la eficiencia que se vayan a evaluar según las necesidades del rol encargado de ejecutar el procedimiento, y la fase del ciclo de vida en el que se encuentre>

2 Identificar métricas

<Se identifican las métricas necesarias para cada sub-característica, en la siguiente tabla se realiza esta actividad>

Sub-característica	Nombre de las métricas escogidas para realizar la evaluación

3 Ejecución de la evaluación

<Se procede a ejecutar la evaluación del procedimiento>

- **Calcular métricas**

<Se realiza el cálculo de las métricas seleccionadas.

En la siguiente tabla se recoge todo el proceso de evaluación>

Sub-característica	Valor	Nivel
Rendimiento		
Utilización de recursos		

Tabla: Ejecución de la evaluación

4 Analizar posibles resultados

<Se analizan los resultados dando las conclusiones de la evaluación efectuada.

Incluir en el plan de mediciones las métricas escogidas para realizar la evaluación.